PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G11B 20/18 — A1

(11) International Publication Number: WO 91/13437

(43) International Publication Date: 5 September 1991 (05.09.91)

(21) International Application Number: PCT/US91/01056

(22) International Filing Date: 19 February 1991 (19.02.91)

(30) Priority data:
481,427 20 February 1990 (20.02.90) US

(71) Applicant: EASTMAN KODAK COMPANY [US/US]; 343 State Street, Rochester, NY 14650-2201 (US).

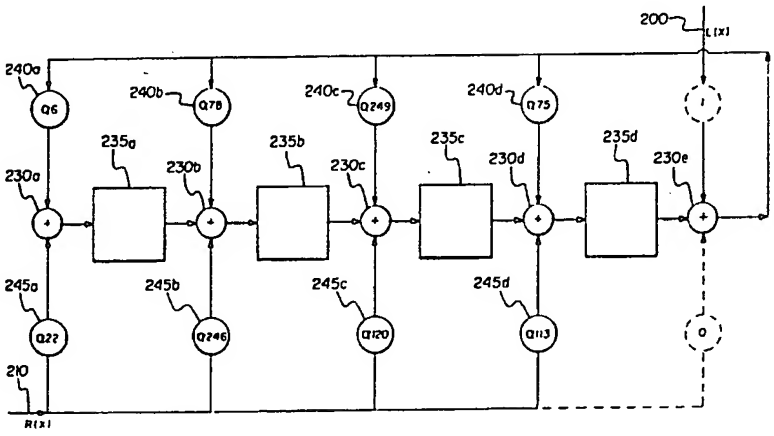
(72) Inventor: WELDON, Edward, J., Jr. ; 1152 Kealaolu Avenue, Honolulu, HI 96816 (US).

(74) Agent: KURZ, Warren, W.; 343 State Street, Rochester, NY 14650-2201 (US).

(81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published
With international search report.

(54) Title: A HIGH SPEED ENCODER FOR NON-SYSTEMATIC CODES



(57) Abstract

Interleaved cyclic error correction encoded binary data, of the type employed in the compact audio disk standard for example, in which the parity characters are located in the middle of a codeword, are encoded by a serial encoder to greatly increase the speed at which such data may be recorded. In the invention, the message characters are treated as two polynomials, a left-hand sub-codeword or message block representing the message characters to the left of the parity characters and a right-hand sub-codeword or message block representing the message characters to the right of the parity characters. The encoder of the invention has two parallel inputs (200, 210) which simultaneously receive the message characters of the right-hand and left-hand sub-codewords, respectively, each input receiving one character at a time in serial fashion. The encoder is a serial register in which each stage receives the sum of the contents of the previous stage and (a) the product of the current character of the left-hand sub-codeword multiplied (245) by a respective coefficient of the code generator polynomial and (b) the product of the current character of the left-hand sub-codeword multiplied by a respective coefficient of the residue of an algebraic shift factor, whose degree is equal to the length of the cyclic error correction code minus the length of the sub-codeword, divided by the code generator polynomial.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TG	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		

A HIGH SPEED ENCODER FOR NON-SYSTEMATIC CODES

BACKGROUND OF THE INVENTION

Technical Field:

The invention is related to error correction encoders useful with interleaved data encoded with an outer cyclic error correction code of the type employed, for example, in the compact audio disk industry standard, in which a parallel encoder is typically required because the format is such that the parity characters are in the middle of the codeword.

Background Art:

In data recording systems, such as the industry standard compact audio disk system, it is necessary to be able to correct not only short random errors in the data but also long bursts of errors occurring therein. A well-known technique is to interleave the data, encode it with an outer error correction code, interleave the outer error correction encoded data, encode the interleaved outer error correction encoded data with an inner error correction code and finally interleave the inner error correction encoded data. Each of the three interleaving steps can perform a different interleaving scheme, as desired. As one example of such a system, the industry standard compact audio disk system employs an outer error correction code in which the check portion, containing the parity characters, is in the middle of the codeword, having message characters on either side of it. The outer code employed is a (28,24) shortened Reed-Solomon code over $GF(2^8)$ in which the first twelve characters in the outer codeword are message characters, the middle

four characters are parity characters and the last twelve characters ar message characters. The inner code in the compact audio disk standard does not have the check portion in the middle of the inner codeword but rather at the end of the codeword.

The advantage of this system is that, while the encoding process in the recording system is relatively time-consuming, both burst errors and random errors are simply and quickly corrected in the decoder during retrieval or playback of the data, as described in U.S. Patent No. 4,413,340 to Odaka et al. and assigned to Sony Corporation. This feature minimizes the cost of a compact audio disk player. The referenced Odaka et al. patent describes the inner and outer error correction encoders as well as the interleaving schemes employed at the input and output of each encoder in the industry standard compact audio disk system.

Typically, such systems are intended for use where the speed of recording need not be particularly fast. Such is the case, for example, in the industry standard compact audio disk system in which digital stereo sound is permanently recorded on a master optical disk by a laser beam. The optical pattern thus produced on the master disk is then easily and quickly reproduced on a large number of disks which are then made available to the public. The only requirement is that the decoder in the compact audio disk player keep up with the playback of the data. It does not matter how long the encoder takes to process the data for recording on the master disk. Accordingly, the encoding system described in the above-referenced Odaka et al. patent is not particularly

fast. In fact, the speed of the encoder is limited by the natur of the outer code chosen for the industry standard compact audio disk syst m.

The recording system of the compact audio disk system as described in the above-referenced Odaka et al. patent employs parallel interleavers and parallel encoders which require the simultaneous presence of all characters to be interleaved at each interleaver and to be encoded at each encoder. Thus, the first interleaver must completely interleave all incoming message characters in an incoming data block before the outer encoder can begin encoding to form an outer codeword, and the second interleaver must finish interleaving the outer codeword before the inner encoder can begin encoding the interleaved outer codeword. Thus, there are inherent delays which are unavoidable in a system employing parallel encoders and parallel interleavers. Such delays would not be present in a serial system which serially interleaves and serially encodes the message characters as they are received one character at a time. Such a serial system cannot function using the parallel encoders described above and must instead use serial encoders in place thereof. Unfortunately, the outer parallel Reed Solomon encoder used in the compact audio disk system cannot be replaced by a serial Reed Solomon encoder, because the serial Reed Solomon encoder will not provide the correct parity characters for the outer codeword.

Specifically, since the outer code requires that the check characters of the outer codeword be located in the middle of the word (i.e., in "intermediate" symbol locations in the codeword), it is not possible to encode

the data in serial fashion using a conventional byte serial encoder in accordanc with the outer code. This is because the value of the parity characters depends in part upon their location within the codeword, and conventional byte-serial encoders can only compute parity characters for placement at the end of the codeword. The degree of the polynomial corresponding to the check characters generated by a serial encoder is lower than the degree of the polynomial representing the message characters, and therefore the check characters generated by a serial encoder correspond to symbol locations at the end of the codeword only. As described, for example, in Peterson, W. W., and E. J. Weldon, Jr., Error-Correcting Codes, Ed. II, MIT Press (1972), all of the message characters are shifted through a serial encoder, and when this is done the serial encoder contains the parity characters which are to be appended at the end of all the message characters to form a codeword.

One way around the apparent inability of a serial encoder to compute parity characters having intermediate symbol locations would be to cyclically shift to the left-most symbol locations all of the message characters located in symbol locations to the right of the parity characters, so that the parity characters would be located at the end of the shifted codeword. However, the code is a shortened cyclic code, as mentioned above. Therefore, the resulting shifted codeword would consist, from left to right, of a group of message characters in the left-most symbol locations, a long string of zero-valued characters in intermediate symbol locations and the remaining message characters immediately followed by the parity characters in the right-most symbol locations.

The length f the string of zero-valued characters is equal to the size of the code minus the number of message and parity characters in the shortened codeword. Since parity characters in the shifted codeword are located at the end of the codeword, a conventional byte-serial shift register encoder can be used to compute them. Unfortunately, this would not necessarily speed up the encoding process as might be expected because of the extra time wasted as the string of zero-valued characters in the middle of the new codeword is shifted through the serial encoder. (For the reader not familiar with algebraic coding theory, it should be noted that the serial encoder in this case must actually shift all of the zero-valued characters through itself in order to arrive at the right answer, and not simply "skip ahead" past the string of zero-valued characters to the next group of message characters to avoid wasting time.) In the case of the industry standard compact audio disk system, the size of the code is 255 and there are 24 message characters and 4 parity characters, so that the length of the string of zeroes in the shifted codeword would be 227. In order to compute the parity characters, the 24 message characters and the 227 zero-valued characters would have to be shifted through the serial encoder, for a total of 251 shifts. The extra 227 shifts (and computations) which must be performed by the serial encoder in such a case would represent an unacceptable delay and is therefore not feasible.

Thus, given the choice of the outer error correction code as a shortened cyclic code having check characters located in intermediate symbol locations, a serial encoder cannot be employed in systems such as the

industry standard compact audio disk system. Accordingly, the delays described above which are inherent in a parallel interl aving and encoding system such as the industry standard compact audio disk system appear to be unavoidable. Therefore, it has seemed that there is no practical way to increase the speed at which data can be recorded in a system employing a code in which the parity characters are in intermediate symbol locations.

DISCLOSURE OF THE INVENTION

An error correction system embodying the invention employs an outer code and an inner code in which the message portion of the inner codewords are interleaved codewords of the outer code and in which the parity characters of the outer code are located at intermediate symbol locations and uses byte-serial encoders and byte-serial interleavers exclusively.

A cyclic error correction code serial encoder embodying the invention encodes codewords in accordance with a code whose codewords have their parity characters located in intermediate symbol locations by performing a number of shifts equal to ONE-HALF the number of message characters in the codeword. In the case of the industry standard compact audio disk system, the number of shifts required by the serial encoder of the invention is only 12. This contrasts dramatically with the expectation described above that the number of shifts required would be 251 in such a case. Thus, the encoding speed for this type of system is increased beyond any expectation in the art.

The serial encoder of the invention has successive serial stages each connected to receive results computed in the previous stage by a Galois field adder. The output of the last stage recirculates back to the input of the first stage. Each of two ("left" and "right") serial inputs of the encoder receives one message character at a time from a respective one of the two groups of message characters which are to be located to the left and right, respectively, of the group of parity characters in the codeword. Each message character received at the "left" input is separately multiplied by each of the coefficients of the code generator polynomial and a respective one of the resulting products is individually received at the adder of a corresponding one of the encoder stages. Each message character received at the "right" input of the encoder is separately multiplied by each of the coefficients of a special "shift" polynomial and each one of the resulting products is individually received at the adder of a corresponding one of the encoder stages. The special shift polynomial, h(x), is the remainder of division by the code generator polynomial of the algebraic shift factor x^n otherwise required in the example given previously to shift to the left-most symbol locations the message characters located to the right of the parity characters in the codeword. This may be stated as follows:

(0) $h(x) = \text{remainder}[x^n/g(x)]$.

In the case of the (28,24) Reed Solomon code of the industry standard compact audio disk system, n = 243 and the code generator polynomial in $GF(2^8)$ is

(1) $g(x) = x^4 + \alpha^{75}x^3 + \alpha^{249}x^2 + \alpha^{78}x + \alpha^6$.

Therefore, in this exemplary case the special shift polynomial employed in the serial encoder of the

invention is

(2) $h(x) = \alpha^{113}x^3 + \alpha^{120}x^2 + \alpha^{246}x + \alpha^{22}$.

The message characters received at the left input is pre-multiplied or shifted by a distance equal to the number of parity characters. This is simply accomplished by connecting the left-hand encoder input to the highest order or last stage of the encoder.

The two groups of message characters are sent to the right-hand and left-hand encoder inputs simultaneously one character at a time. As soon as all of the message characters have been shifted through all of the encoder stages, each stage contains one of the parity characters.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is described in detail below by reference to the accompanying drawings, of which:

Fig. 1a is a diagram of a codeword whose parity check characters are located in the middle of the codeword;

Fig. 1b is a diagram of a byte-serial Reed Solomon encoder of the prior art;

Fig. 1c is a diagram of a shifted version of the codeword of Fig. 1a in which the parity check characters are located on the right end of the codeword and which can be encoded by the encoder of Fig. 1b;

Fig. 2 is a simplified block diagram of a serial Reed Solomon encoder embodying the invention;

Fig. 3 is a block diagram of an error correction system embodying the invention;

Fig. 4 is a block of the encoder of Fig. 2 as employed in the system of Fig. 3; and

Fig. 5 is a simplified block diagram of a typical scrambling element employed in each interleaver in the system of Fig. 3.

MODES FOR CARRYING OUT THE INVENTION

A shortened (28,24) Reed Solomon code over $GF(2^8)$ has codewords of the form illustrated in Fig. 1a. This is the code employed in the industry standard compact disk system. The codeword of Fig. 1a is remarkable in that its four parity check characters 110 are located in the middle of the codeword between two groups of twelve message characters 100 and 120, rather than at the right end of the codeword. Each character is a binary eight-bit byte representing one of the elements of $GF(2^8)$. Such a codeword may be thought of as representing a polynomial in which the coefficients of the terms x^0 through x^{11} are the first twelve message characters 100, the coefficients of the terms x^{12} through x^{15} are the four parity check characters 110 and the coefficients of the terms x^{16} through x^{27} are the last twelve message characters 120.

As described previously herein, the parity check characters 110 of the codeword cannot be computed by a conventional byte-serial encoder, but instead must be computed using matrix methods which require all of the message characters to be processed together in parallel,

in accordance with the referenced patent to Odaka et al. In order to compute the parity check characters, a serial enc der requires that the parity check character locations are the lowest order symbol locations, at the right end of the codeword, corresponding to the coefficients of x^0, x^1, x^2, x^3, etc. Such a serial encoder simply divides the polynomial corresponding to the message characters by the code generator polynomial and the remainder of the division contains the parity check characters and has the structure illustrated in Fig. 1b.

Referring to Fig. 1b, a serial encoder consists of a series of stages each including a register 130 and a Galois field adder 135, which is an exclusive OR gate. The input to each stage is connected to the output of the previous stage, the output of the last stage being fed back to each of the adders 135 through a respective multiplier 140. Each multiplier 140 multiplies the value fed back from the last stage by a respective one of the coefficients of the code generator polynomial. The encoder of Fig. 1b encodes in accordance with the code generator polynomial of Equation (1) above. All of the elements shown in Fig. 1b are eight bits wide. Each message character is received one at a time at the adder 135e. Each register 130 is strobed in synchronism with the receipt of each new message character at the adder 135e to shift the contents of the registers 130 from left to right. In essence, the encoder divides a polynomial whose coefficients are the incoming message characters by the code generator polynomial and stores the coefficients of the remainder (not the quotient) in the registers 130. The quotient is simply discarded. After all of the

message characters have been shifted through the encoder of Fig. 1b, each of the coefficients stored in the registers 130 is a corresponding one of the parity check characters to be appended to the end of the codeword.

The codeword of Fig. 1a can be transformed into a form which is encodable by the encoder of Fig. 1b. For this purpose, the message characters 100 to the right of the parity characters 110 are designated as a right-hand polynomial R(x) while the message characters 120 to the left of the parity characters 110 are designated as a left-hand polynomial L(x). Using the RS(28,24) code as an example, if the codeword of Fig. 1a is shifted to the right by multiplication of the polynomials corresponding to the characters 100, 110, 120 by an algebraic factor x^{243}, the codeword is transformed to the form illustrated in Fig. 1c, in which the right-most characters 100 now appear at the left end of the codeword and are separated from the next group of characters 120 by a long string of zero-valued characters 125. The shifted codeword of Fig. 1c is a valid codeword because any shift of a codeword of a cyclic code is another codeword. The codeword of Fig. 1a corresponds to the codeword polynomial

(3) $C(x) = x^{16}L(x) + x^{12}P(x) + R(x)$,

where P(x) is the polynomial of parity check characters, while the shifted codeword of Fig. 1c corresponds to the codeword polynomial

(4) $C'(x) = x^{243}R(x) + x^{4}L(x) + P(x)$.

In essence, R(x) (represented by the twelve message characters 100) consisted of twelve coefficients of x^0 through x^{11} in Fig 1a and now, by virtue of being multiplied by x^{243}, consists of twelve coefficients of x^{243} through x^{254} in Fig. 1c. Of course, the length of the

codeword has now been increased from 28 to 255. As a result, while the encoder of Fig. 1b can now compute the parity check bytes as the message bytes 120, 125 and 100 are shifted through the encoder in order, an additional 227 shifts of the encoder is required, entailing an unacceptable delay in the encoding process. By way of comparison, it should be noted that if the parity check characters 110 had been located at the right end of the codeword of Fig. 1a rather than in the middle, the encoder of Fig. 1b could have computed the parity check characters with only twenty-four shifts. An object of the invention is to provide a byte-serial encoder which encodes the codeword of Fig. 1c without the extra 227 shifts.

The serial encoder of the invention is illustrated in Fig. 2. One input 200 of the encoder of Fig. 2 receives the coefficients of the polynomial L(x) while the other input 210 receives the coefficients of the polynomial R(x). The input 200 is coupled to four serial stages in the manner of the serial encoder of Fig. 1b. Each of the stages of the serial encoder of Fig. 2 consists of an eight-bit Galois field adder 230 and an eight-bit register 235. As in Fig. 1b, each of the adders 230a through 230d receives feedback from the last adder 230e through a respective multiplier 240 which multiplies the feedback value by a respective one of the code generator polynomial coefficients.

Each coefficient of the right-hand polynomial received at the second encoder input 210 is multiplied in an individual multiplier 245 by a respective one of the coefficients of a special shift polynomial h(x). The

special shift polynomial is the remainder of the division by the codeword generator polynomial by the algebraic shift factor which shifted R(x) to the left end of the codeword of Fig. 1c. In the specific example discussed above, the algebraic shift factor was x^{243}. As discussed above in connection with equations (1) and (2), the special shift polynomial h(x) is given in equation (2). The multipliers 245a through 245d multiply each character of R(x) by the coefficients of h(x) α^{22}, α^{246}, α^{120} and α^{113}, respectively, as indicated in Fig. 2.

The twelve coefficients of R(x) and the twelve coefficients of L(x) are fed simultaneously to the respective one of the inputs 210, 200 one character at a time. After the encoder of Fig. 2 has been shifted twelve times so that all of the coefficients of R(x) and L(x) have been shifted into the encoder, each of the shift registers 235 contains one of the four parity check characters to be appended to the shifted codeword of Fig. 1c.

It can be shown that the multiplication of R(x) by h(x) replaces the multiplication of R(x) by x^{243} in equation (4). The multiplication of L(x) by x^4 in equation (4) is accomplished by simply adding the contents of the highest order stage (the adder 230e) to L(x) (at the input 200) prior to its multiplication by the g(x) multipliers 240a-d of Fig. 2, while R(x) (at the input 210) is coupled directly to the h(x) multipliers 245a-d and thence to the adders 230a-d of the four lowest order stages.

In essence, the encoder of Fig. 2 is a linear

feedback register which multiplies R(x) by h(x), multiplies L(x) by x^4, divides the sum of the resulting products by the g(x) (the code generator polynomial) and stores the remainder.

The multiplication of R(x) by h(x) can be justified as follows: It has been pointed out in the above-referenced text by Peterson and Weldon at page 368 that in decoding a codeword having a long string of zeroes of length s, simplification is achieved by multiplying the received codeword by the remainder of $x^s/g(x)$ prior to decoding. A similar approach can be stated for the encoder of Fig. 2. The shifting of the long string of 227 zeroes through the encoder can be dispensed with if right-hand message character polynomial R(x) is multiplied by the remainder of $x^{243}/g(x)$, which is h(x). Multiplication by h(x) is equivalent to multiplication by x^{243} because

$$\text{(5)}\quad \text{remainder}\{x^{243}R(x)/g(x)\} = \text{remainder}\{[\text{remainder}(x^{243}/g(x))]/g(x)\}.$$

However, from the definition of h(x), the expression on the right side of equation (5) is

$$\text{remainder}\{h(x)\ R(x)/g(x)\}.$$

The product h(x) R(x) is produced at the output of the multipliers 245a-d and the division by g(x) is performed as the output of each multiplier 254 is shifted through the registers 235 in the encoder of Fig. 2.

Conveniently, the number of multipliers 245 required to represent the coefficients of h(x) is the same as the number of multipliers 240 required to represent the coefficients of g(x). This is true not only in the case of the choice of code generator polynomial for the

compact audio disk system discussed her in but for any cyclic code whose parity check characters are in intermediate symbol locations. The coefficients of g(x) are preferably normalized so that the leading coefficient (of the highest power of x) is unity, as in equation (1) above. Thus, L(x) at the left input 200 is coupled to the highest order adder 230e through a wire--corresponding to a multiplication by unity. This reduces the number of multipliers 240 required to represent coefficients of g(x) to four, the degree of g(x). The degree of h(x) is always one less than g(x), so that the number of multipliers representing the coefficients of g(x) and h(x) is the same in the preferred embodiment of the invention. In the embodiment Fig. 2, the coefficient of x^4 in h(x) is 0. The coefficient of 0 is simply implemented in Fig. 2 by the lack of any connection between the adder 230e corresponding to x^4 and R(x) (at the input 210).

SERIAL INTERLEAVING AND ENCODING SYSTEM

The serial encoder of Fig. 2 enables the serial encoding and processing of codewords whose parity characters are in intermediate symbol locations. For example, the interleaving and encoding sub-system of the industry standard compact audio disk recording system can be transformed from the parallel design disclosed in the above-referenced patent to Odaka et al. to a serial byte processing system illustrated in Fig. 3. The serial system of Fig. 3 permits the recording of data at least 9.8 times faster than the parallel system disclosed in the Odaka et al. patent.

Referring now to Fig. 3, the serial interleaving and

encoding system generates data for recording in accordance with the data format described in the Odaka et al. patent and required in the industry standard compact audio disk system. The system includes a first serial interleaver 300 which interleaves incoming user data in accordance with the crossover and delay pattern of the parallel interleaver 1 of Fig. 1A of U.S. Patent No. 4,413,340 to Odaka et al. However, unlike the parallel interleaver of the Odaka et al. patent, the serial interleaver 300 receives and outputs data one byte at a time. Each block of twenty-four user bytes is received at the input 300a of the serial interleaver 300 of Fig. 3 one byte at a time beginning with the input byte depicted at the top of the left (input) side of the interleaver 1 of Fig. 1A of the Odaka et al. patent and ending with the byte depicted at the bottom thereof. The serial interleaver 300 transmits at its output 300b block of twenty-four interleaved bytes on at a time beginning with the output byte depicted at the top of the right (output) side of the interleaver 1 of Fig. 1A of the Odaka et al. patent and concluding with the byte depicted at the bottom thereof. The structure of the serial interleaver 300 of Fig. 3 will be described below.

An outer serial encoder 310 is the serial encoder of Fig. 2. The serial encoder 310 produces the parity check characters for a codeword of the type illustrated in Fig. 1a. The serial encoder 310 receives from the serial interleaver 300 a block of twenty-four interleaved bytes one byte at a time and passes these bytes along at its output 310a to a second serial interleaver 320. It then transmits the four parity check characters at its output 310a. In the Odaka et al. patent, this function

is performed in parallel fashion by the coder 8 of Fig. 1A thereof. The detailed design of the serial encoder 310 for the system of Fig. 3 is described below.

The second serial interleaver 320 interleaves the 28-byte codewords received from the outer serial encoder 310 in accordance with the crossover and delay pattern of the parallel interleaver 9 of Fig. 1B of U.S. Patent No. 4,413,340 to Odaka et al. However, unlike the parallel interleaver of the Odaka et al. patent, the serial interleaver 320 receives and outputs data one byte at a time. Each codeword of twenty-eight bytes is received at the input 320a of the serial interleaver 320 of Fig. 3 one byte at a time beginning with the input byte depicted at the top of the left (input) side of the interleaver 9 of Fig. 1B of the Odaka et al. patent and ending with the byte depicted at the bottom thereof. The serial interleaver 320 transmits at its output 320b a block of twenty-eight interleaved bytes one at a time beginning with the output byte depicted at the top of the right (output) side of the interleaver 9 of Fig. 1B of the Odaka et al. patent and concluding with the byte depicted at the bottom thereof. The structure of the serial interleaver 320 of Fig. 3 will be described below.

An inner serial encoder 330 receives at its input 330a the interleaved outer codewords from the second serial interleaver 320. The inner serial encoder 330 provides the function performed in parallel fashion by the coder 10 of Fig. 1B of U.S. Patent No. 4,413,340 to Odaka et al. This function is to encode each incoming 28-byte block in accordance with a systematic code in which the parity check characters are located at the end

of the codeword. A conventional serial encoder suffices to perform this function, as there are no parity characters in intermediate symbol locations in such a systemic code. Accordingly, in the serial encoding system of Fig. 3, the serial encoder 330 is a conventional Reed-Solomon serial encoder of the type well-known in the prior art and which is described for example in the above-referenced text by W. W. Peterson and E. J. Weldon, Jr. The conventional serial encoder 330 employs the Reed-Solomon code described in connection with the coder 10 of Fig. 1B of the above-referenced patent to Odaka et al. The twenty-eight bytes of each block to be encoded are received one at a time at the input 330a of the serial encoder 330. They are transmitted at the encoder output 330b, followed immediately by four parity check characters computed by the serial encoder 330, one byte at a time.

A third serial interleaver 340 interleaves the 32-byte codewords received from the inner serial encoder 330 in accordance with the crossover and delay pattern of the parallel interleaver 11 of Fig. 1B of U.S. Patent No. 4,413,340 to Odaka et al. However, unlike the parallel interleaver of the Odaka et al. patent, the serial interleaver 340 receives and outputs data one byte at a time. Each codeword of thirty-two bytes is received at the input 340a of the serial interleaver 340 of Fig. 3 one byte at a time beginning with the input byte depicted at the top of the left (input) side of the interleaver 11 of Fig. 1B of the Odaka et al. patent and ending with the byte depicted at the bottom thereof. The serial interleaver 340 transmits at its output 340b a block of thirty-two interleaved bytes one byte at a time beginning

with the output byte depicted at the top of th right (output) side of the interleaver 11 of Fig. 1B of the Odaka et al. patent and concluding with the byte depicted at the bottom thereof. The structure of the serial interleaver 340 of Fig. 3 will be described below.

The output 340b of the third serial interleaver is connected to an 8-to-14 bit channel encoder 350, which produces data processed by recording circuits not shown in Fig. 3.

The preferred embodiment of the two-input outer serial encoder 310 is illustrated in Fig. 4. It is an implementation of the generic two-input encoder illustrated in Fig. 2. All data lines are eight bit wide, as noted in Fig. 4, while all control and clocking lines are one bit wide. Each adder 230 of Fig. 2 is implemented by an 8-bit exclusive OR gate 430 in Fig. 4. Each register 235 of Fig. 2 is implemented by an 8-bit flip-flop register 435 in Fig. 4. Each multiplier 240 of Fig. 2 is implemented by a read-only memory (ROM) 440 in Fig. 4 whose 8-bit address input is the multiplier input and whose 8-bit data output is the multiplier output. Each ROM 440 is programmed in the well-known manner such that the 8-bit data output is the Galois field product of the eight bit input with the corresponding coefficient of the code generator polynomial g(x). Each multiplier 245 of Fig. 2 is implemented by a read-only memory (ROM) 445 in Fig. 4 whose 8-bit address input is the multiplier input and whose 8-bit data output is the multiplier output. Each ROM 445 is programmed in the well-known manner such that the 8-bit data output is the Galois field product of the 8-bit input with the corresponding

coefficient of the special shift polynomial h(x).

Alternate ones of the incoming bytes from the first serial interleaver output 300b are treated as the coefficients of the left-hand message polynomial L(x) while the remaining ones are treated as the coefficients of the right-hand message polynomial R(x). For this purpose, the first interleaver output 300b (which constitutes the input to the encoder of Fig. 4) is connected to two input registers 450R and 450L, which control the incoming bytes associated with the polynomials R(x) and L(x), respectively. The incoming bytes are stored alternately in the input registers 450L and 450R in synchronism with clock signals CLK L IN and CLK R IN, respectively. After each byte of L(x) is received in the input register 450L, it is applied to the "left" encoder input 400 connected to the highest order adder 430e. It is also applied through an eight-bit tri-state buffer 460 to the encoder output (connected to the second interleaver input 310) in synchronism with a clock signal CLK L OUT. Similarly, after each byte of R(x) is received in the input register 450R, it is applied to the "right" encoder input 410 as well as the encoder output by the register 450R in synchronism with a clock signal CLK R OUT. The left encoder input 400 is connected to the inputs of the g(x) multiplier ROM's 440 while the right encoder input is connected to the inputs of the h(x) multiplier ROM's 445. As soon as an L(x) byte has been sent to the left input 400 and an R(x) byte has been sent to the right input 410, all of the registers 435a-d are clocked once so as to shift bytes through the encoder left-to-right by one stage in synchronism with a clock signal CLK SHIFT. The foregoing operation is repeated

twelve times (in the case of the Reed-Solomon (28,24) code discussed above in connection with Fig. 1a) so that all of the L(x) and R(x) bytes are shifted into the encoder of Fig. 4. Then, outputs of all of the multiplier ROM's 440 and 445 are disabled by a clock signal CLK OUT so that no multiplication takes place in the encoder and the contents of the registers 435a-d are shifted left-to-right in synchronism with the clock signal CLK SHIFT through a tri-state output buffer 470 to the encoder output. The tri-state encoder output buffer is enabled for this purpose by the clock signal CLK OUT.

The portion of the encoder of Fig. 4 which is enclosed in dashed line including the adders 430, the registers 435 and the tri-state output buffers 460 and 470, was implemented in a programmable logic array PAL 16R4A-4CNSTD in accordance with standard practice.

Each of the serial interleavers was implemented using a generic structure corresponding to that illustrated in Fig. 5. The interleaving is accomplished by temporarily storing the incoming bytes in individual memory locations in a random access memory (RAM) 500 in a first order and then reading them out in a second order which is scrambled in a predetermined manner with respect to the first order. The scrambling is defined in accordance with the respective interleaving schemes discussed previously herein and illustrated in the above-referenced Odaka et al. patent, by programming a ROM 510 connected to the address input 500a of the RAM 500. A clock generator 520 with which the bytes are synchronized drives an 8-bit up-counter 530 whose output is connected to the address input of the ROM 510. As bytes are

received at the interleav r, the up-counter 530 sends a sequence of monotonically increasing count bytes to the ROM 510 as it counts up in synchronism with the clock generator 520. Each count byte stimulates the ROM 510 to send a unique address byte to the RAM 500 specifying a unique address for writing the current incoming data byte. The progression of address bytes generated by the ROM 510 is defined differently for each of the serial interleavers 300, 320 and 340 and is scrambled with respect to the monotonic progression of count bytes generated by the up-counter 530. After all of the bytes of the current block have been written into the RAM 500, control of the RAM's address input is taken over by the up-counter 530, and the bytes are read out of the RAM 500 in a monotonically increasing order of memory addresses. For this purpose, the RAM address input is connected to the outputs of the ROM 510 and the up-counter 530 through 8-bit gates 540 and 550 respectively. The gates 540 and 550 are controlled by clock signals CLK WRITE and CLK READ, respectively, which are respectively enabling during the writing and the reading of the data in the RAM 500.

The first interleaver 300 of Fig. 3 was implemented with a version of the structure of Fig. 5 in which the ROM 510 is programmed in accordance with the table of Appendix A. For each address byte generated by the ROM 510, Appendix A lists a set of counts received from the up-counter 530 which stimulate transmission of that address byte. The second interleaver 320 of Fig. 3 was implemented with a version of the structure of Fig. 5 in which the ROM 510 is programmed in accordance with the table of Appendix B. For each address byte transmitted

by the ROM 510, Appendix B lists a set of counts received from the up-counter 530 which stimulate transmission of that address byte. The third interleaver 340 was implemented with a version of the structure of Fig. 5 in which the ROM 510 was programmed in accordance with the table of Appendix C. For each address byte transmitted by the ROM 510, Appendix C lists a set of counts received from the up-counter which stimulate transmission of that address byte.

While the invention has been described in connection with a non-systemic RS(28,24) code having parity character symbol locations in the middle of the codeword, other embodiments are useful with trivial modifications with other non-systematic codes whose parity characters have various intermediate symbol locations.

While the invention has been described in detail by specific reference to preferred embodiments thereof, it is understood that variations and modifications thereof may be made without departing from the true spirit and scope of the invention.

What is claimed is:

1. A serial shift encoder for generating parity characters having intermediate symbol locations between first and second message blocks in a codeword of a code generator polynomial g(x), comprising:

first means for generating first products of a successive character of the first message block multiplied by corresponding coefficients of g(x);

second means for generating second products of a successive character of the second message block multiplied by corresponding coefficients of a remainder h(x) of the division by g(x) of an algebraic shift factor x^n wherein the degree n of said algebraic shift factor x^n is equal to the length of the cyclic error correction code minus the length of said right-hand message block; and

third means comprising a series of adders for generating in a corresponding one of said adders the sum of (a) the contents of a previous one of said adders, (b) a respective one of said first products and (c) a respective one of said second products, and serially shifting the sum generated in each adder to a succeeding adder in said series.

2. The encoder of Claim 1 wherein the products summed in successive ones of said adders correspond to coefficients of g(x) and h(x) of successively higher powers of x.

3. The encoder of Claim 2 wherein said first and second message blocks are the higher and lower order coefficients, respectively, of a codeword polynomial, said encoder further comprising:

means for multiplying said successive character of said first message block by a second algebraic shift factor x^m, where m equals to the number

of parity character locations between said first and second message blocks in said codeword.

4. The encoder of Claim 3 wherein said means for multiplying said successive character of said first message block by said second shift factor comprises means for adding to said successive character of said first message block the contents of an adder in said series of adders to the highest degree of x.

5. The encoder of Claim 4 wherein the highest order coefficient of g(x) corresponing to said highest order adder is unity and said respective coefficient of h(x) corresponding to said highest order adder is zero.

6. The encoder of Claim 5 wherein said successive characters of said first message block are transmitted directly to said highest order adder means without multiplication and said successive characters of said second message block are not coupled to said highest order adder means.

7. A method for generating parity characters having intermediate symbol locations between first and second message blocks in a codeword of a code generator polynomial g(x), comprising:

first generating first products of a successive character of the first message block multiplied by corresponding coefficients of g(x);

second generating second products of a successive character of the second message block multiplied by corresponding coefficients of a remainder h(x) of the division by g(x) of an algebraic shift factor x^n, wherein the degree n of said algebraic shift factor x^n is equal to the length of the cyclic error correction code minus the length of said right-hand message block;

third generating in a corresponding one of a series of adders the sum of (a) the contents of a previous one of said adders, (b) a respective one of said first products and (c) a respective one of said second products; and

serially shifting the sum generated in each adder to a succeeding adder in said series.

8. The method of Claim 7 wherein the products summed in successive ones of said adders correspond to coefficients of g(x) and h(x) of successively higher powers of x.

9. The encoder of Claim 7 wherein said first and second message blocks are the higher and lower order coefficients, respectively, of a codeword polynomial, said encoder further comprising:

means for multiplying said successive character of said first message block by a second algebraic shift factor x^m, where m equals to the number of parity character locations between said first and second message blocks in said codeword.

10. The method of Claim 9 wherein said step of multiplying said successive character of said first message block by said second shift factor comprises adding to said successive character of said first message block the contents of an adder in said series of adders to the highest degree of x.

11. The method of Claim 10 wherein the highest order coefficient of g(x) corresponding to said highest order adder is unity and said respective coefficient of h(x) corresponding to said highest order adder is zero.

12. The method of Claim 11 wherein said first generating step comprises transmitting the characters of said first message block directly to said highest

order adder means without multiplication and wherein said second generating step comprises isolating said highest order adder means from the characters of said second message block.

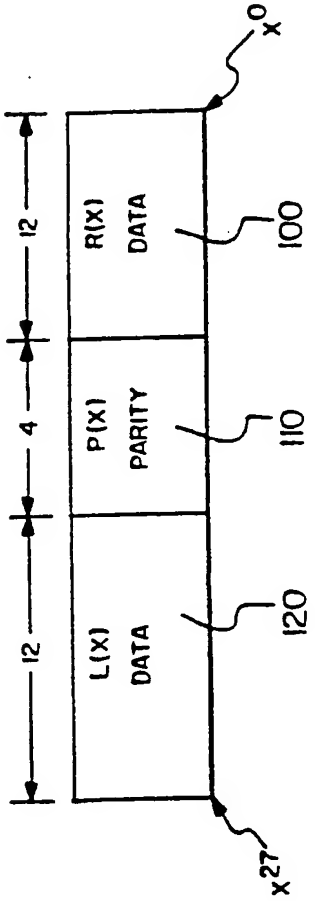

x0
R(X)
DATA
100
P(X)
PARITY
110
L(X)
DATA
120
x27
12
4
12

FIG. Ia

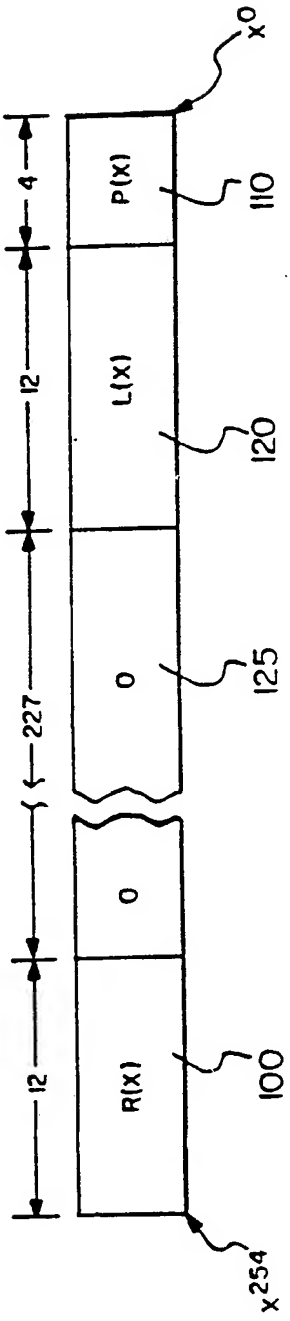

x0
P(X)
110
L(X)
120
0
125
0
R(X)
100
x254
4
12
227
12

FIG. Ic

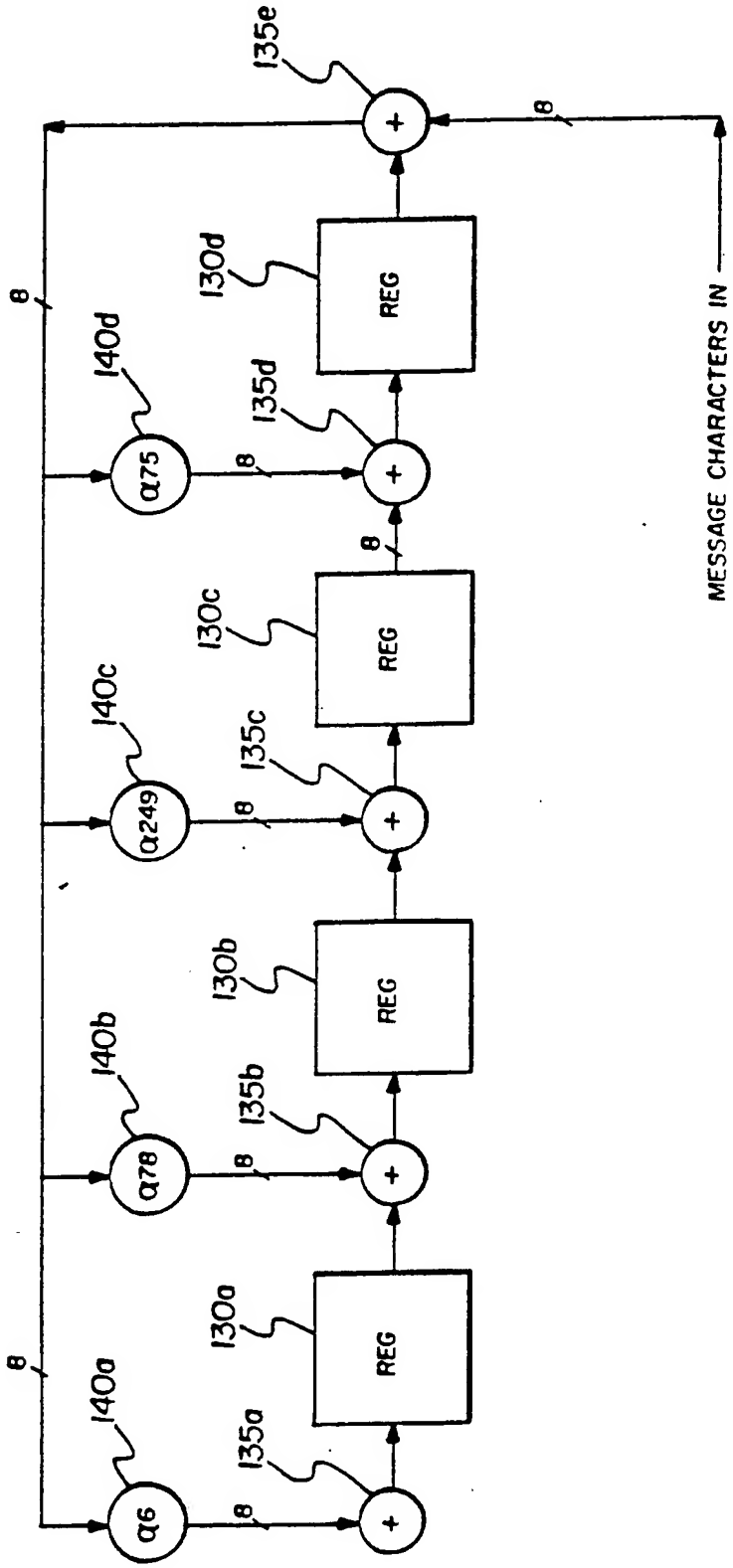


FIG. 1b
"PRIOR ART"

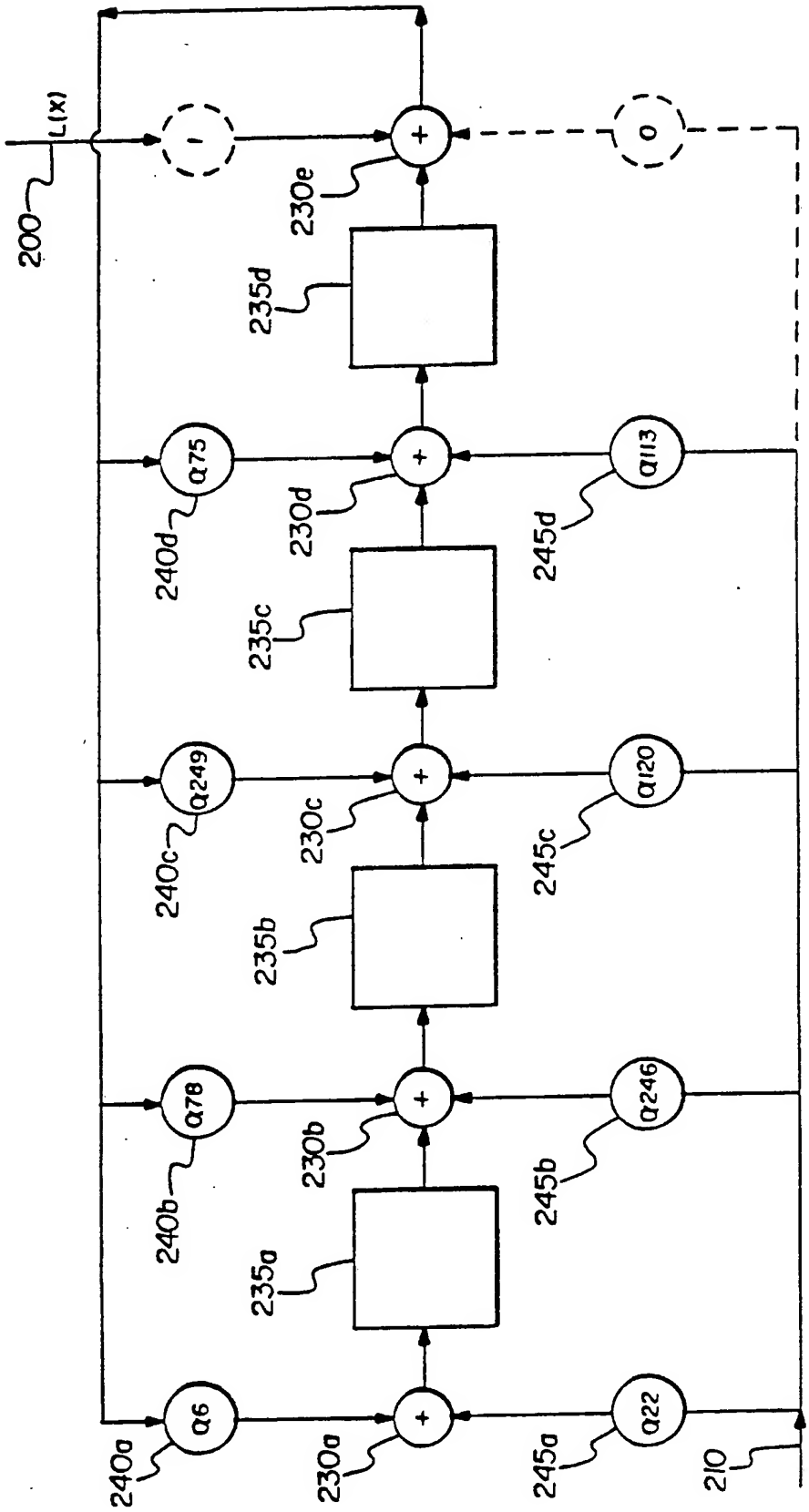

L(x)
200
-
+
0
230e
235d
Q75
240d
230d
235c
Q113
245d
Q249
240c
230c
235b
Q120
245c
Q78
240b
230b
235a
Q246
245b
Q6
240a
230a
Q22
245a
210
R(x)

FIG. 2

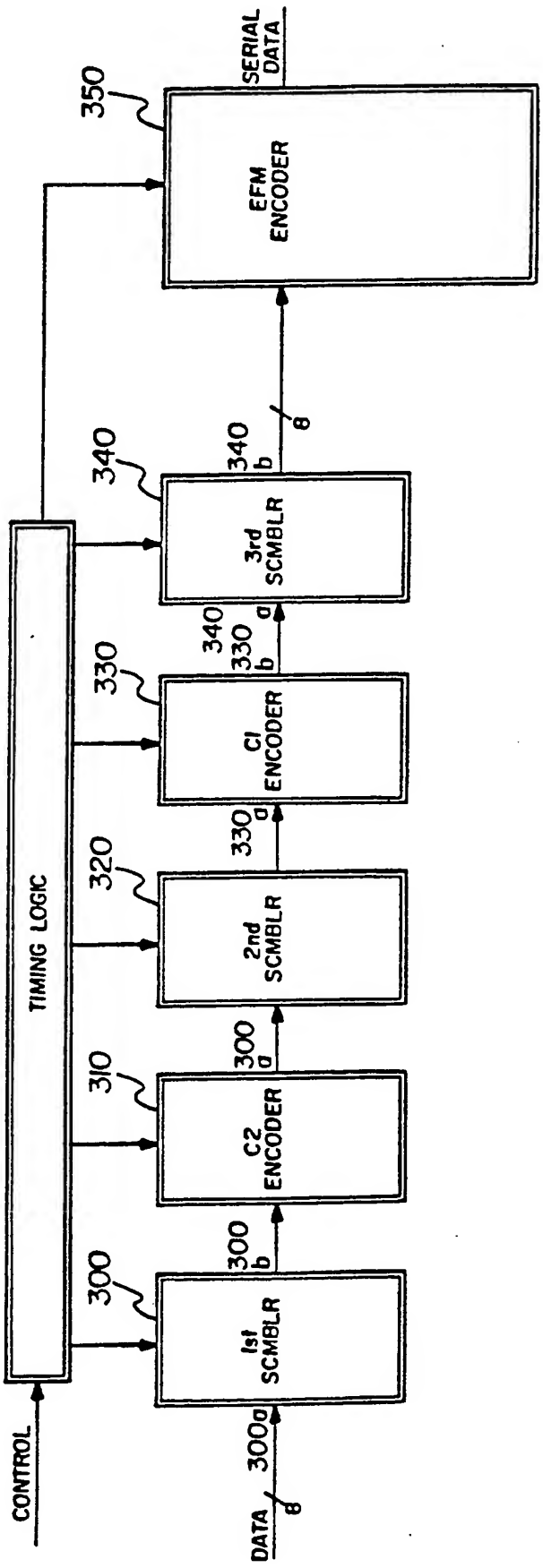

SERIAL DATA
350
EFM ENCODER
8
340 b
340
3rd SCMBLR
340 b
330 d
330
C1 ENCODER
330 d
320
2nd SCMBLR
300 d
310
C2 ENCODER
300 b
300
1st SCMBLR
300d
8
DATA
CONTROL
TIMING LOGIC

FIG. 3

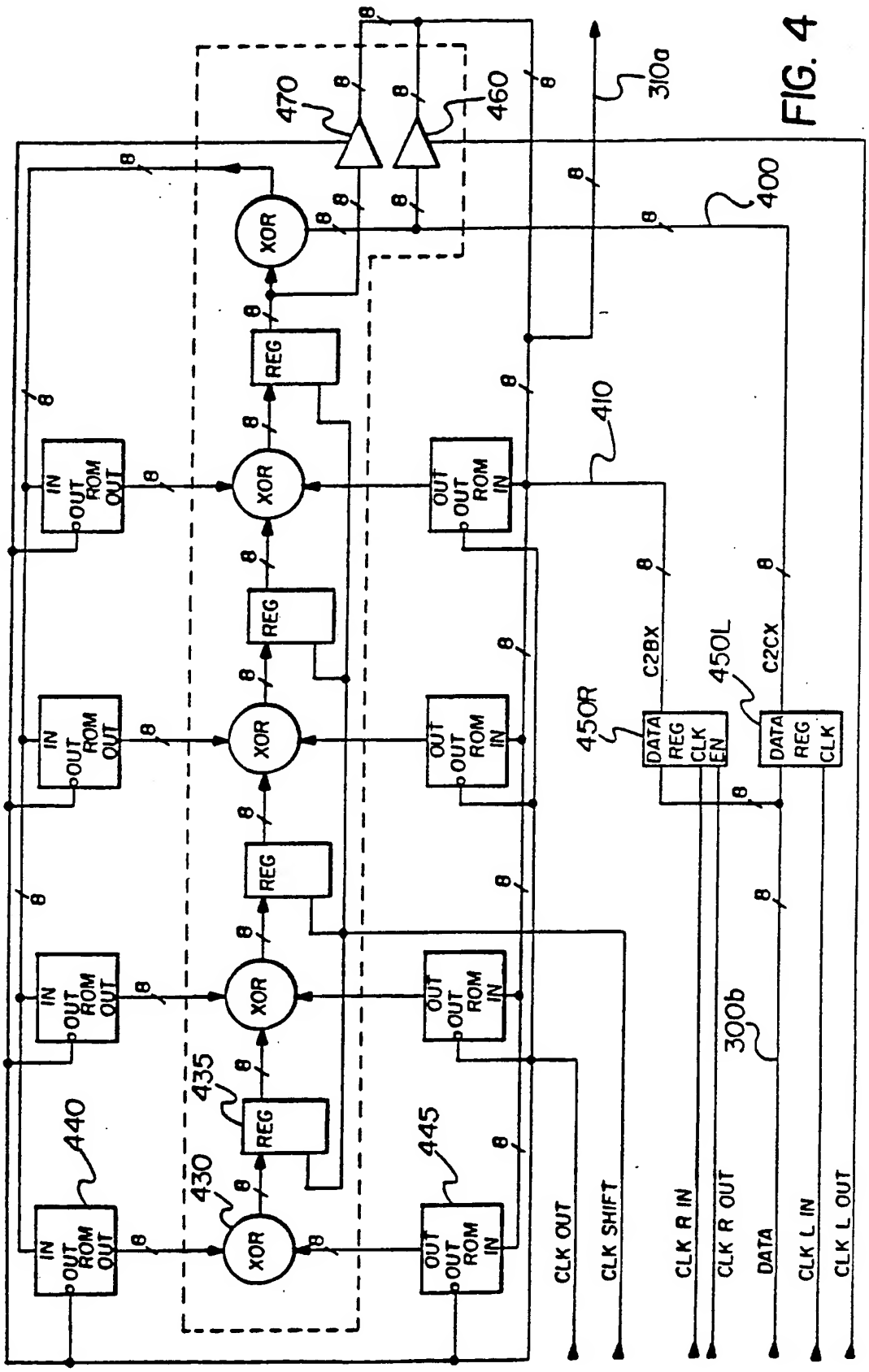

470
460
310a
400
410
XOR
REG
IN
OUT
ROM
OUT
OUT
OUT
ROM
IN
C2BX
C2CX
450L
450R
DATA
REG
CLK
EN
DATA
REG
CLK
435
440
430
445
300b
CLK OUT
CLK SHIFT
CLK R IN
CLK R OUT
DATA
CLK L IN
CLK L OUT

FIG. 4

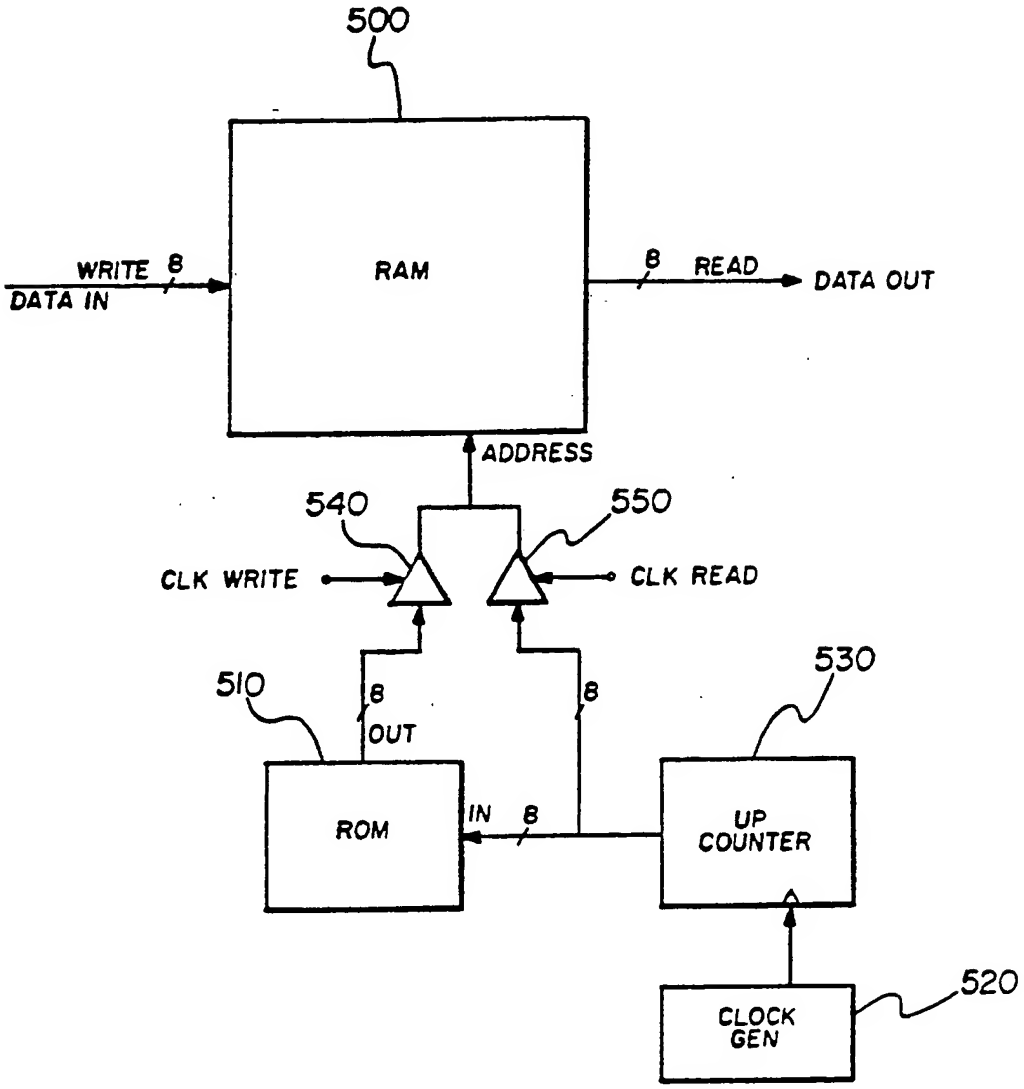

500
RAM
WRITE 8
DATA IN
8 READ
DATA OUT
ADDRESS
540
550
CLK WRITE
CLK READ
510
8
OUT
8
530
ROM
IN 8
UP
COUNTER
520
CLOCK
GEN

FIG. 5

INTERNATIONAL SEARCH REPORT

International Application No PCT/US 91/01056

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [6]

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC[5]: G 11 B 20/18

II. FIELDS SEARCHED

Minimum Documentation Searched [7]

Classification System	Classification Symbols
IPC[5]	G 11 B 20/00,H 03 M 13/00

Documentation Searched other than Minimum Documentation to the Extent that such Documents are included in the Fields Searched [8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [9]

Category [10]	Citation of Document, [11] with indication, where appropriate, of the relevant passages [12]	Relevant to Claim No. [13]
A	EP, A2, 0 291 167 (MATSUSHITO ELECTRIC INDUSTRIAL) 17 November 1988 (17.11.88), see abstract, page 4, lines 1-26, claim 1. --	1,7
P,A	EP, A2, 0 364 172 (ADVANCED MICRO DEVICEC, INC) 18 April 1990 (18.04.90), see abstract; page 2, lines 26-36; claims 1,4,11,14,21; fig. 2. --	1,2,3, 7,8,9
A	EP, A2, 0 136 587 (KABUSHIKI KAISHA TOSHIBA) 10 April 1985 (10.04.85), see abstract; fig. 2; claims. --	1;7
A	EP, A1, 0 188 627 (SONY CORPORATION) 30 July 1986 (30.07.86), see abstract; fig. 10; claim	1,7

* Special categories of cited documents: [10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
24 May 1991	10 JUN 1991
International Searching Authority	**Signature of Authorized Officer**
EUROPEAN PATENT OFFICE	MISS T. TAZELAAR

Form PCT/ISA/210 (second sheet) (January 1985)

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category *	Citation of Document, 11 with indication, where appropriate, of the relevant passages	Relevant to Claim No.
	1.	
A	US, A, 4 189 710 (IGA) 19 February 1980 (19.02.80), see abstract; fig. 2,3; column 3, lines 6-21.	1,7
A	IEE PROCEEDINGS-I, Volume 136, Part I, Number 1, February 1989 (London, GB), S.D. Bate et al. "Error control techniques applicable to HF channels, pages 57-63, see totality.	1,7

ANHANG zum internationalen Recherchenbericht über die internationale Patentanmeldung Nr.

ANNEX to the International Search Report to the International Patent Application No.

ANNEXE au rapport de recherche international relatif á la demande de brevet international n°

PCT/US91/01056 SAE 45315

In diesem Anhang sind die Mitglieder der Patentfamilien der im obengenannten internationalen Recherchenbericht angeführten Patentdokumente angegeben. Diese Angaben dienen nur zur Unterrichtung und erfolgen ohne Gewähr.

This Annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The Office is in no way liable for these particulars which are given merely for the purpose of information.

La présente annexe indique les membres de la famille de brevets relatifs aux documents de brevets cités dans le rapport de recherche international visée ci-dessus. Les renseignements fournis sont donnés á titre indicatif et n'engagent pas la responsibilité de l'Office.

Im Recherchenbericht angeführtes Patentdokument Patent document cited in search report Document de brevet cité dans le rapport de recherche	Datum der Veröffentlichung Publication date Date de publication	Mitglied(er) der Patentfamilie Patent family member(s) Membre(s) de la famille de brevets	Datum der Veröffentlichung Publication date Date de publication
EP-A2- 291167	17-11-88	EP-A3- 291167	05-09-90
		JP-A2-63257966	25-10-88
		US-A - 4949342	14-08-90
		JP-A2-63255876	24-10-88
EP-A2- 364172	18-04-90	JP-A2- 2211723	23-08-90
EP-A2- 136587	10-04-85	DE-C0- 3484455	23-05-91
		EP-A3- 136587	27-05-87
		EP-B1- 136587	17-04-91
		JP-A2-60055733	01-04-85
		US-A - 4608692	26-08-86
		JP-A2-60055565	30-03-85
EP-A1- 188627	30-07-86	WO-A1- 8601026	13-02-86
		DE-C0- 3580101	15-11-90
		EP-B1- 188627	10-10-90
		JP-A2-61029461	10-02-86
		US-A - 4852101	25-07-89
		JP-A2-61029460	10-02-86
US-A - 4189710	19-02-80	AT-A - 3608/78	15-04-82
		AT-B - 369211	10-12-82
		CA-A1- 1098212	24-03-81
		DE-A1- 2821772	30-11-78
		FR-A1- 2391605	15-12-78
		FR-B1- 2391605	27-01-84
		GB-A - 1601638	04-11-81
		JP-A2-53142105	11-12-78
		NL-A - 7805385	21-11-78